Draft

(1)

ADST/WDL/TR--93--003064

AD-A284 469

ADST
Software Maintenance Manual
for the
BDS-D VIDS-equipped M1

Loral Western Development Labs
Electronic Defense Systems Software Department
Software Engineering Laboratory
3200 Zanker Road
P.O. Box 49041
San Jose, CA 95161-9041

July 11, 1993

Contract No. N61339-91-D0001
CDRL No. A002

Prepared for

Simulation Training and Instrumentation Command
Naval Training Systems Center
12350 Research Parkway
Orlando, FL 328266-3275

LORAL

REPORT DOCUMENTATION PAGE

Form approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	11 July 1993	

4. TITLE AND SUBTITLE
ADST Software Maintenance Manual for the BDS-D VIDS-equipped M1.

5. FUNDING NUMBERS
Contract No. N61339-91-D-0001

6. AUTHOR(S)
Stivers, Douglas

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Loral Systems Company
ADST Program Office
12443 Research Parkway, Suite 303
Orlando, FL 32826

8. PERFORMING ORGANIZATION REPORT NUMBER
ADST/WDL/TR-93-003064

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Simulation, Training and Instrumentation Command
STRICOM
Naval Training Systems Center
12350 Research Parkway
Orlando, FL 32826-3275

10. SPONSORING ORGANIZATION REPORT

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (Maximum 200 words)

This document provides an overview of the hardware and software used to build and execute the Vehicle Integrated Defense System (VIDS) equipped M1.

14. SUBJECT TERMS

15. NUMBER OF PAGES
8

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	17. SECURITY CLASSIFICATION OF THIS PAGE	17. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE of CONTENTS

LIST of FIGURES

1. Introduction

1.1. Scope

This document provides an overview of the hardware and software used to build and simulate the Vehicle Integrated Defense System (VIDS) equipped M1. The VIDS-equipped M1 provides a suite of sensors and countermeasures designed to increase the survivability of an M1 tank crew.

1.2. Purpose

This document provides information required to successfully build the VIDS-equipped M1 software. Diagrams and descriptions are provided to support coldstarts and software enhancements.

1.3. Reference documents

1.3.1. Government documents.

SPECIFICATIONS:

1. PM-Survivability: VIDS Armored Vehicle Survivability Equipment (AVSE) BDS-D Functional Specifications, 29 May 1992.

1.3.2. Non-Government documents.

1. Loral: Software Design Document for the VIDS-Equipped M1 Tank Simulator of BDS-D, Contract No. N61339-91-D0001, July 14, 1993

2. Loral: Battlefield Distributed Simulation-Development (BDS-D) Vehicle Integrated Defense System (VIDS) Feasibility Analysis Report, 14 October 1992.

3. ADST Coldstart Procedures Manual for the BDS-D VIDS-equipped M1, Contract No. N61339-91-D0001, July 1, 1993.

4. BBN: The SIMNET Network and Protocols, Report 7627, June 1991.

2. Overview

The VIDS-equipped M1 provides sensors and countermeasures designed to increase the survivability of an M1 tank crew against smart enemy weapons. Reports from the sensors are used to alert the crew, prioritize threats and select appropriate countermeasures.

2.1. Architectural Overview

VIDS simulates a new capability for the M1 tank simulator. In general, VIDS executes as one of several major components of the M1 simulator. Consequently, other entities within the simulated battlefield cannot distinguish a VIDS-equipped tank from any other tank.

VIDS is composed of two independently executing applications: one application executes on a GT simulating sensors, the threat resolution module and countermeasures, while the other application executes on a PC simulating the user interface and visual display as well as activating audible alerts heard on the tank intercom. Like other entities within the simulated battlefield, the two VIDS applications communicate with one another using digital messages transmitted on an Ethernet® local area network. Messages conform to the format described in "The SIMNET Network and Protocols" and are referred to as Protocol Data Units (PDUs).

3. Hardware Configuration

3.1. Hardware Description

No hardware changes have been made to the GT. On the other hand, the PC is completely new. The PC is configured as follows:

a. Everex *Tempo* 386/33
b. 4MB of RAM
c. 52 MB Quantum hard disk.
d. 13" Sony Monitor.
e. Elographics E274 Touch Screen
f. Racal Interlan NI5210-16 Ethernet card.
g. Creative Labs Inc., Soundblaster card.

Finally, the PC runs under DOS 5.0.

3.1.2. Custom Hardware

None.

3.2. Hardware to Software Interfaces

During PC bootup, the AUTOEXEC.BAT executes \SYS\ETHERNET\NI5210, a terminate-and-stay-resident (TSR) assembly language program. It is this program from the Free Software Foundation which handles the low-level hardware interfaces to the NI5210 card.

During the VIDS-equipped M1 initialization, the VIDS PDU is added to the list of other multicast messages. The multicast message list defines which network messages are retrieved by the GT during the simulated battle. All other messages are discarded.

4. Software Description

4.1. VIDS Software Description

The detailed design of the VIDS software can be found in reference 1 of the Non-Government References documents in section 1.3.2.

4.2. Development Environment Descriptions

Since VIDS executes on two host computers, each development environment will be described separately.

4.2.1. GT Development Environment Directory Structure

The GT software is written in C. Standard UNIX® (SunOS release 4.0.3) utilities running on a SUN 3 workstation are used to build the application.

Major components of the GT software have been organized into a directory hierarchy of software libraries, refer to Figures 1 and 2. To support software reuse, the GT software is partitioned into two major subdirectories: common and simnet. Both common and simnet represent roots of other subdirectory hierarchies.

The majority of the VIDS software resides in simnet/vehicle/libsrc/libvids. This includes the software for the simulated sensors, countermeasures and threat resolution module. Software for reading VIDS PDUs is located in simnet/vehicle/libsrc/libRcvNet while software for writing VIDS PDUs is located in simnet/vehicle/libsrc/libSendNet

```
common
  makefile
  tools
  include
  lib
  libsrc
    libappidc
    libassoc
    libbbd
    libchannel
    libcif
    libclparse
    libdtad
    libex
    libfifo
    libfilter
    libhash
    libidc
    libiv
    libkeybrd
    liblist
    liblogger
    libmatrix
    libmem
    libmoves
    libnetif
    libp2p
    libparser
    libpvis
    librtc
    librva
    libshm
    libsv
    libsvdvr
    libtdb
    libtmr
    libtty
    libuseful
  data
```

Figure 1. Directory Hierarchy for GT-resident common Subdirectory

```
simnet
  makefile
  tools
  include
  lib
  libsrc
    libapp
    libcmc
    libcontrols
    libdev
    libevent
    libimage
    libimpacts
    libio_simul
    libiod
    liblrf
    libmap
    libmath
    libmatrox
    libnet_simul
    libpfile
    libpots
    libquat
    librva_util
    libser
    libsoftp
    libsound
    libterrain
    libtimers
    libtrack
    libutil
  vehicle
    include
    lib
    libsrc
      libRcvNet
      libSendNet
      libaero
      libair
      libball
      libbigwh
      libcig
      libcloud
      libdyn
      libfail
      libfield
      libgeoball
      libground
      libhet
      libhull
      libkin
      libmain
      libmissile
      libmsg
      libmun
      libnear
      libnewkin
      libobjects
      libproc
      librotate
      libsad
      libsmoke
      libspaceball
      libsusp
      libturret
      libupdate
      libveh
      libvflags
      libvids
    vids
      include
      src
  bin
```

Figure 2. Directory Hierarchy for GT-resident simnet Subdirectory

Makefiles for common and simnet exist to perform all operations necessary to perform partial or full software coldstarts. These Makefiles are found within respective tools directories. Note that the coldstart for common must precede the coldstart for simnet.

During a build, individual source files are compiled and copied into a local library file which is then copied into higher level library (lib) directory. Once all the libraries have been rebuilt or updated, the application is linked into an executable file as the last step within the simnet makefile.

To rebuild the software within the common subdirectory, establish common as the current working directory and type the following commands:

```
make clean
make headers
make install
```

The first command, "make clean", recursively deletes all object and library files within the directory tree. The second command, " make headers", recursively copies local header files to a centralized include directory. The third command, "make install", recursively compiles individual source files, builds local libraries of compiled objects and copies these libraries to a centralized lib directory.

To rebuild the software within the simnet subdirectory, establish simnet as the current working directory and type the same sequence of commands used to build the common subdirectory.

The resulting executable file (simnet/vehicle/vids/src/vids) must be copied to a tape cartridge. The tape cartridge is then used to copy the executable file to the target GT hard disk.

4.2.2. PC Development Environment Directory Structure

The majority of the PC software is written in C++, only two low-level functions are written in assembly. The Borland C++ software development environment (version 3.1) is used to edit/compile and link the application.

The directory hierarchy of software is significantly simpler than the GT software, refer to Figure 3.

```
vids
   vidscom
   sendpkt
   sound
```

Figure 3. Directory Hierarchy for PC-resident Software

The Borland environment utilizes the concept of a project file to establish the same type of software dependencies used by the UNIX make utility. A menu option under **Compile** allows both partial and full software coldstarts.

4.3. Runtime Environment Description

The VIDS runtime environment is described in ADST Coldstart Procedures Manual for the BDS-D VIDS-equipped M1. A new file, VIDS.D, resides on the GT within the following subdirectory: /simnet/vehicle/vids/data. The contents of this file define the available sensors and countermeasures for a given simulated battle. Figure 4 lists a subset of its content. Omission of MWS or LWR parameters deactivates the corresponding sensor; omission of ROS or MCD parameters deactivates the corresponding countermeasures. However, when a countermeasure is deactivated, the corresponding button mapping must be deactivated (set to NULL_JAM_Switch or NULL_Salvo_Switch) so that the corresponding button on the PC display screen is appears as SPARE.

4.4. Startup Procedure

The VIDS startup procedure is described in ADST Coldstart Procedures Manual for the BDS-D VIDS-equipped M1.

5. Utility Software

None.

6. Notes

List of Acronyms:

ADST	Advanced Distributed Simulation Technology
BDS-D	Battlefield Distributed Simulation - Developmental
GT	Graphics Technology
PC	Personal Computer
TSR	Terminate and Stay Resident
VIDS	Vehicle Integrated Defense System

```
#
MWS.Response_Time_in_sec                        1.2
MWS.Threat_Priority                             2
MWS.Alarm_Index                                 1
MWS.Alarm_Duration_in_sec                       3.0
MWS.Max_Detection_Distance_in_meter             6000.0
MWS.Max_Approach_Angle_in_Deg                   22.5
MWS.Azimuth_Coverage_Central_Angle_in_Deg       0.0
MWS.Azimuth_Coverage_Delta_in_Deg               180.0
MWS.Elevation_Coverage_Central_Angle_in_Deg     15.0
MWS.Elevation_Coverage_Delta_in_Deg             25.0
MWS.Detection_Probability_MWS                   0.98
MWS.Detection_Accuracy_in_Deg                   2.0
MWS.Life_Countdown_in_sec                       30.0
#
LWR.Response_Time_in_sec                        0.5
LWR.Threat_Priority                             1
LWR.Alarm_Index                                 1
LWR.Alarm_Duration_in_sec                       3.0
LWR.Azimuth_Coverage_Central_Angle_in_Deg       0.0
LWR.Azimuth_Coverage_Delta_in_Deg               180.0
LWR.Elevation_Coverage_Central_Angle_in_Deg     15.0
LWR.Elevation_Coverage_Delta_in_Deg             25.0
LWR.Detection_Probability_LRF                   0.92
LWR.Detection_Probability_LBR_LDES              0.97
LWR.Detection_Accuracy_in_Deg                   3.0
LWR.Life_Countdown_in_sec                       30.0
#
# Valid values are MCD_JAM_Switch or NULL_JAM_Switch
Jam_Button_Map                              MCD_JAM_Switch
#
# Valid values are ROS_Salvo_Switch or NULL_Salvo_Switch
Salvo_Button_Map                            ROS_Salvo_Switch
#
ROS.Coverage_Angle_in_Deg                       15.0
ROS.Max_Turret_Rotation_Rate                    45.0
ROS.Launch_Distance_in_meter                    30.0
ROS.Response_Time_in_sec                        2.0
#
MCD.Response_Time_in_sec                        0.2
MCD.Jam_Time_in_sec                             3.0
MCD.Azimuth_in_Deg                              22.0
MCD.Elevation_in_Deg                            5.0
MCD.Max_Turret_Rate                             45.0
#
```

Figure 4. Key Elements of VIDS.D